COMP6258 - A REPRODUCIBILITY STUDY ON SPMI

Gaurang Chitnis
University of Southampton
gccc1n24@soton.ac.uk

Jiayi Fang
University of Southampton
jf4n24@soton.ac.uk

Yijia Deng
University of Southampton
yd8u24@soton.ac.uk

Yaru Qian
University of Southampton
yq4n22@soton.ac.uk

1 INTRODUCTION

Reproducibility in machine learning has become an area of significant concern. Studies show that a large number of published results cannot be reproduced independently Baker (2016); Semmelrock et al. (2023). Specifically, according to the survey Baker (2016), More than 70% of researchers have tried and failed to reproduce another scientist's experiments, and more than half cannot reproduce their own. Semi-supervised partial label learning (SSPLL) is particularly challenging as methods must handle both label redundancy and insufficiency. This report investigates the reproducibility of the work by Liu et al. Liu et al. (2024). The paper proposes SPMI (Semi-supervised Partial label learning with Mutual Information) which is a novel framework that claims to uniformly handle both partial labelled and unlabelled data through a Mutual Information (MI) based approach. The paper shows this by claiming superior performance over the baselines across multiple datasets.

1.1 REPRODUCIBILITY AIMS

Since the original paper does not provide any code, we will be implementing SPMI and the other methods that are used as a baseline from scratch by following the methodology provided in the paper. The code base with the result CSV and logs can be found in this GitHub repo: https://github.com/COMP6258-Reproducibility-Challenge/SPMI-Reproducibility-Study

We aim to reproduce the following: 1) SPMI's superior performance over baselines on Fashion-MNIST, CIFAR-10, and SVHN, 2) Effectiveness of label expansion/condensation mechanisms, and 3) The Method's ability to maintain low error rates while purifying candidate label sets.

These claims made by Liu et al. are particularly significant as they challenge the conventional "divide and conquer" methods of SSPLL and propose a unified framework for handling mixed supervision scenarios. As such, in this report, we shall not only present the results of our reproducibility study but also shed some light on the practicality and efficiency of SPMI.

2 IMPLEMENTATION DETAILS

We implemented SPMI from scratch based on the equations, algorithm 1, and the experimental details outlined in Liu et al. (2024). The implementation closely follows the paper's specifications, with the label expansion mechanism in Section 3.3, the label condensation strategy in Section 3.4, and the uniform treatment framework in Section 3.2.

2.1 CORE ALGORITHM IMPLEMENTATION

The SPMI algorithm was implemented as described in Algorithm 1 of the original paper. The label channel facilitates dynamic exchange within candidate label sets. Our implementation follows the three component structure: initialisation for unlabelled data, label expansion, and label condensation. The initialisation phase implements Equation 3, where for each unlabelled instance, we add label k to the candidate set if $f_k(x_i) > 1/c$. This ensures that the output for the true label exceeds random probability. A key implementation detail not specified in the paper was handling edge cases where no labels meet the threshold criterion. We did deviate a bit from this by adding a fallback mechanism that selects the top 2 highest probability labels to ensure that there are no empty candidate sets. This was done so that we don't receive an empty list. Since the original paper does not provide algorithms for the other tasks, we shall do so instead.

Algorithm 1: Initialize Unlabelled Candidates

```
Data: model, dataloader, threshold
foreach batch in dataloader do
    $P \leftarrow$ softmax($model(images)$);
    for $i \leftarrow 1$ to batch_size do
        if is_labelled$_i$ then continue;
        $C \leftarrow \{k : P_{i,k} >$ threshold$\}$;
        if $|C| = 0$ then
            $C \leftarrow$ top_k($P_i, 2$)
        update_candidate_set($i, C$);
```

2.2 Label Expansion and Condensation

The label expansion mechanism implements the mutual information from Equation 11, where label k is added when $q(k|z_i) > p(k)$. This directly follows the paper's observation that mutual information $I(Z, k) > 0$ indicates relevance. For labelled data, expansion is constrained to the original candidate set as specified in line 16 of Algorithm 1. The only difference between our implementation and what the paper has given is that we vectorise the operation. This approach is ultimately just more efficient and shouldn't affect the SPMI algorithm. Label condensation follows Equation 15, computing the information score $G(x_i, S_i, k)$ using KL divergence to measure information loss when removing label k. The paper's mathematical description omitted crucial implementation details regarding probability renormalisation. Our implementation ensures numerical stability through the handling of zero probabilities and proper renormalisation after candidate removal.

Algorithm 2: Expand Labels

```
Data: outputs, masks, is_labelled, priors, original_masks
Result: Updated masks
$P \leftarrow$ softmax($outputs$);
$U \leftarrow$ clone($masks$);
for $i \leftarrow 1$ to batch_size do
    for $k \leftarrow 1$ to num_classes do
        if $masks_{i,k} = 1$ then continue;
        if $P_{i,k} > priors_k$ then
            if is_labelled$_i$ $\wedge$ $original\_masks_{i,k} = 1$ then
                $U_{i,k} \leftarrow 1$
            else $U_{i,k} \leftarrow 1$;
return $U$;
```

2.3 Loss Function and Class Prior Updates

Implements Equations 1-2 with weighted negative log-likelihood over candidate sets. Class priors are updated via Equation 16 with EMA where $\mu^{(t+1)} = \alpha\mu^t + (1 - \alpha)\mu^t$. The paper specifies $\alpha = 1.0$ (no smoothing), which is implemented exactly as described.

2.4 Experimental Setup

We followed the experimental setup detailed in Appendix A.4 of the paper. We used batch size 256, SGD with momentum 0.9, weight decay 5×10^{-4}. The architectures used were: LeNet (Fashion-MNIST), WideResNet-28-2 (CIFAR-10/SVHN), WideResNet-28-8 (CIFAR-100). The thresholds

Algorithm 3: Condense Labels

```
Data: outputs, masks, is_labelled, tau
Result: Condensed masks
$P \leftarrow \mathrm{softmax}(outputs)$;
for $i \leftarrow 1$ to batch_size do
    $C \leftarrow \{k : masks_{i,k} = 1\}$;
    if $|C| \leq 1$ then continue;
    foreach $k \in C$ do
        $R \leftarrow C \setminus \{k\}$;
        $P' \leftarrow \mathrm{normalize}(P_{i,R})$, $Q' \leftarrow \mathrm{normalize}(P_{i,C\setminus\{k\}})$;
        $G_k \leftarrow \mathrm{KL\_divergence}(Q', P')$;
    $(G_{\max}, k_{\max}) \leftarrow \max(G\_scores)$, $(G_{\min}, k_{\min}) \leftarrow \min(G\_scores)$;
    $\theta \leftarrow \begin{cases} tau, & \text{if } is\_labelled_i \\ unlabelled\_tau, & \text{otherwise} \end{cases}$;
    if $G_{\max} > \theta$ then
        $masks_{i,k_{\min}} \leftarrow 0$
return $masks$;
```

params were: $\tau = 3$ (partial), $\tau = 2$ (unlabelled). Learning rate selection from 0.05, 0.03, 0.01 with cosine scheduling following the paper's methodology with warm-up epochs T_w from 5, 10, 20.

2.5 Data Augmentation Implementation

We implemented the Weak-strong strategy, where weak augmentation includes random horizontal flipping and random cropping and the strong augmentation additionally incorporates AutoAugment and Cutout. AutoAugment was excluded for Fashion-MNIST as per the paper. EMA updates were used for candidate sets (partial data) and unlabelled data (CIFAR-100).

2.6 Baseline Implementation

The PRODEN+FixMatch baseline was implemented following the approach described in Section A.3. PRODEN provides consistent classification risk estimation through progressive identification and FixMatch contributes consistency regularisation with weak-strong augmentation pairs. Our implementation includes vectorised soft label updates and optimised KL divergence computations for computational efficiency.

2.7 Key Implementation Differences and Challenges

Several implementation choices required careful consideration due to ambiguities in the paper that ultimately affected our attempt at reproducing the paper. For class prior updates (Equation 16), the paper specifies $\alpha = 1.0$. In EMA notation where $\mu^{(t+1)} = \alpha \cdot \mu^t + (1 - \alpha) \cdot \mu^{new}$, setting $\alpha = 1.0$ results in $\mu^{(t+1)} = \mu^{new}$ a complete replacement with no historical smoothing. We initialise priors to uniform values (1/C) and implement this direct replacement approach exactly as specified. The paper's label generation condition $q(k|z_i) > p(k)$ requires estimating class priors $p(k)$, but provides no details on initial estimation or updates during training. We initialise class priors uniformly (1/C) and update them each epoch using Equation 16, making $p(k)$ equivalent to μ_k. This dynamic updating ensures priors adapt to the evolving pseudo-label distribution during training. Label condensation's information score $G(x_i, S_i, k)$ requires probability renormalisation after removing candidates, but the paper omits numerical stability details. We added epsilon-based smoothing (1e-10) to prevent NaN values. A major point of confusion was the Information Bottleneck penalty from Section 3.3. While deriving SPMI using IB principles (Equation 4), the paper never states whether the β penalty should be included in the final loss. We implemented it optionally but found the main experiments use $\beta = 0$, treating IB as purely theoretical motivation. The paper mentions maintaining candidate sets within $[1, c - 1]$ but omits details on how these constraints are enforced in practice. Our implementation naturally prevents empty sets through the initialisation fallback mechanism and condensation that only removes when multiple candidates exist. The upper bound $(c-1)$ is inherently satisfied since true labels aren't added to their own candidate sets. The paper provides a range of learning rate values but doesn't specify which lr works the best for what setup. We used lr of 0.03 for CIFAR10, SVHN and 0.01 for FMNIST.

3 Results and Discussion

We successfully implemented the core SPMI algorithm and achieved partial reproduction on Fashion-MNIST, CIFAR-10, and SVHN.

Performance Results:

SPMI Implementation (Single-GPU): Our best single GPU results achieved 68.62% accuracy on FMNIST (l=4000, p=0.3) vs 89.82% reported. CIFAR-10 (l=4000, p=0.3) got 61.10% vs 92.81. FMNIST (l=1000, 0.3, lr = 0.01) got 62.14% vs 86.47%. Moreover, with a learning rate of 0.03(included by the author) FMNIST (l=1000, 0.3) stagnated at 10% accuracy with no change in average unlabelled candidates per epoch. These results show that SPMI works but requires more careful hyperparameter tuning, mainly with learning rate values outside of the ones provided by the paper. Due to the time and cost we were unable to further test this.

SPMI Implementation (Multi-GPU): Complete failure across all datasets, achieving only random accuracy (10%) due to fundamental DataParallel incompatibility. SVHN was particularly the lowest as it is inherently more challenging than the FMNIST and CIFAR-10 datasets. We shall elaborate on this in the section on Multi-GPU incompatibility.

Baseline Performance: Our baselines significantly outperformed SPMI implementations. FIXMATCH+PRODEN achieved 90.34% on FMNIST (l=4000, p=0.3), 92.57% on Fashion-MNIST (l=1000, p=0.3), 82.08% on CIFAR-10 (l=4000, p=0.3), 94.36% on SVHN (l=1000, p=0.3), and 64.57% on CIFAR-10 (l=1000, p=0.3). Furthermore, POP+FIXMATCH achieved: 90.0±0.2% on Fashion-MNIST (l=1000, p=0.3) and 70.0±0.25% on CIFAR-10 (l=4000, p=0.3). These baselines consistently exceeded our SPMI implementation and were close to the reported accuracies.

Critical Multi-GPU Incompatibility: Despite the paper citing that they used multiple ”NVIDIA RTX 3090 GPUs”, our multi-GPU implementation consistently failed (10% random accuracy). Root cause analysis revealed SPMI's fundamental incompatibility with PyTorch's DataParallel. Where, batch splitting across GPUs means that the candidate set updates only occur on the master GPU using partial outputs. This breaks SPMI's core mechanism. Class prior updates also see incomplete data due to distributed batches. Single-GPU experiments succeeded, confirming this technical limitation. This however was discovered towards the end and as such we have limited experiments run on the ”right” setup.

Invalidated Ablation Study: Our ablation experiments yielded uniformly poor results (17-19% accuracy) due to being conducted on the broken multi-GPU setup before discovering the DataParallel incompatibility. As such, these ablation results are meaningless for assessing component importance. Time constraints prevented rerunning ablation studies on a single GPU after discovering this critical flaw.

Learning Rate Sensitivity: The paper lists learning rates 0.05, 0.03, 0.01 without dataset guidance. We found critical sensitivity; FMNIST required lr=0.01 (lr=0.03 caused complete failure at 10% accuracy and 4 average candidates stagnation). This represents a major reproducibility barrier through undocumented hyperparameter selection.

Computational Cost: Experiments required 12-13 hours per a 500-epoch run on a dual RTX 4090 setup and much longer on a laptop RTX 4080. This computational cost stems from the iterative candidate set updates and mutual information calculations performed at each training step.

Reproducibility Verdict: SPMI is partially reproducible with moderate to high effort on a powerful single GPU. Core algorithm functionality is demonstrated, but performance gaps and architectural limitations indicate missing implementation details. The uniform treatment framework represents a valid contribution, but requires careful dataset-specific hyperparameter tuning and figuring out which component should be used in the SPMI architecture due to some ambiguities. Furthermore, given that PRODEN+FixMatch achieved within 6% of SPMI's reported accuracy while being substantially faster to train, researchers should carefully consider whether SPMI's marginal improvements justify its computational overhead for their specific use cases.

References

Monya Baker. 1,500 scientists lift the lid on reproducibility, 2016.

Yangfan Liu, Jiaqi Lv, Xin Geng, and Ning Xu. Learning with partial-label and unlabeled data: A uniform treatment for supervision redundancy and insufficiency. In *Forty-first International Conference on Machine Learning*, 2024.

Harald Semmelrock, Simone Kopeinik, Dieter Theiler, Tony Ross-Hellauer, and Dominik Kowald. Reproducibility in machine learning-driven research. *arXiv preprint arXiv:2307.10320*, 2023.